IDA PAPER P-2648

COURSEWARE PORTABILITY

J. D. Fletcher

August 1992

Prepared for
Assistant Secretary of Defense
(Force Management and Personnel)
and
Office of the Assistant Secretary of Defense for Public Affairs
(Armed Forces Information Service)
Defense Audiovisual Policy Office

Approved for public release; distribution unlimited.

INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311-1772

IDA Log No. HQ 91-40319

DEFINITIONS

IDA publishes the following documents to report the results of its work.

Reports

Reports are the most authoritative and most carefully considered products IDA publishes. They normally embody results of major projects which (a) have a direct bearing on decisions affecting major programs, (b) address issues of significant concern to the Executive Branch, the Congress and/or the public, or (c) address issues that have significant economic implications. IDA Reports are reviewed by outside panels of experts to ensure their high quality and relevance to the problems studied, and they are released by the President of IDA.

Group Reports

Group Reports record the findings and results of IDA established working groups and panels composed of senior individuals addressing major issues which otherwise would be the subject of an IDA Report. IDA Group Reports are reviewed by the senior individuals responsible for the project and others as selected by IDA to ensure their high quality and relevance to the problems studied, and are released by the President of IDA.

Papers

Papers, also authoritative and carefully considered products of IDA, address studies that are narrower in scope than those covered in Reports. IDA Papers are reviewed to ensure that they meet the high standards expected of refereed papers in professional journals or formal Agency reports.

Documents

IDA Documents are used for the convenience of the sponsors or the analysts (a) to record substantive work done in quick reaction studies, (b) to record the proceedings of conferences and meetings, (c) to make available preliminary and tentative results of analyses, (d) to record data developed in the course of an investigation, or (e) to forward *information that is essentially unanalyzed and unevaluated. The review of IDA Documents* is suited to their content and intended use.

The work reported in this document was conducted under contract MDA 903 89 C 0003 for the Department of Defense. The publication of this IDA document does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
August 1992

3. REPORT TYPE AND DATES COVERED
Final--January-June 1992

4. TITLE AND SUBTITLE
Courseware Portability

5. FUNDING NUMBERS
C - MDA 903 89 C 0003
T- T-L2-883

6. AUTHOR(S)
J. D. Fletcher

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses
1801 N. Beauregard St.
Alexandria, VA 22311-1772

8. PERFORMING ORGANIZATION REPORT NUMBER
IDA Paper P-2648

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
OASD (FM&P)(R&R)
The Pentagon, Room 3B930
Washington, DC 20301

OASD (PA)
American Forces Information
ServiceDefense Audiovisual Policy Office
601 North Fairfax Street
Alexandria, VA 22314-2007

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

"Portability" enables interactive courseware (ICW) and associated application programs to operate on computer-based systems other than the ones on which they are developed. It will increase sharing of ICW across a range of instructional settings within military Services, and across internationally allied military Services. The Department of Defense (DoD) has advocated and implemented a standard for ICW development employing a virtual device interface that currently incorporates specified commands organized into service groups for: (1) system management, (2) visual management, (3) videodisc control, and (4) XY-input devices. This approach, now specified in MIL-STD-1379D and DoD Instruction 1322.20, provides for system-level courseware portability. Future work will expand the standard to: (1) encompass more varieties of ICW, (2) address portability at the device level, (3) address new technological opportunities, (4) better address graphics, (5) encompass more operating systems, and (6) progress from platform independence to authoring software independence. The DoD portability initiative will lower the per-unit costs of ICW, lower instructional system development costs, increase use of advanced instructional technology in military settings, and increase instructional efficiency in the military Services.

14. SUBJECT TERMS
Training, Interactive Courseware, Computer-Based Instruction,Software Portability, Adaptive Training

15. NUMBER OF PAGES
45

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

IDA PAPER P-2648

COURSEWARE PORTABILITY

J. D. Fletcher

August 1992

INSTITUTE FOR DEFENSE ANALYSES
Contract MDA 903 89 C 0003
Task T-L2-883

ACKNOWLEDGMENT

Both the content and the expression of the views in this paper were substantially improved by comments provided by Paul Chatelier, Christopher Dede, Scott Lewis, and Philip Dodds, to all of whom the author is grateful. Errors of fact and exposition remain the sole responsibility of the author.

This work was sponsored by the Office of the Assistant Secretary of Defense for Force Management and Personnel and by the American Forces Information Service, Office of the Assistant Secretary of Defense for Public Affairs. Special thanks are extended to Col. Darwin Anderson, USAF, and James Ashcraft of the Defense Audiovisual Policy Office, Office of the Assistant Secretary of Defense for Public Affairs, and to Gary Boycan of the Training Policy Office, Office of the Assistant Secretary of Defense for Force Management and Personnel, for their guidance, suggestions, and encouragement.

ABSTRACT

"Portability" enables interactive courseware (ICW) and associated application programs to operate on computer-based systems other than the ones on which they are developed. It will increase sharing of ICW across a range of instructional settings within military Services, across military Services, and across internationally allied military Services. The Department of Defense (DoD) has advocated and implemented a standard for ICW development employing a virtual device interface that currently incorporates specified commands organized into service groups for: (1) system management, (2) visual management, (3) videodisc control, and (4) XY-input devices. This approach, now specified in MIL-STD-1379D and DoD Instruction 1322.20, provides for system-level courseware portability. Future work will expand the standard to: (1) encompass more varieties of ICW, (2) address portability at the device level, (3) address new technological opportunities, (4) better address graphics, (5) encompass more operating systems, and (6) progress from platform independence to authoring software independence. The DoD portability initiative will lower the per-unit costs of ICW, lower instructional system development costs, increase use of advanced instructional technology in military settings, and increase instructional efficiency in the military Services.

CONTENTS

FIGURES

TABLES

SUMMARY

A. INTRODUCTION

This paper documents work accomplished thus far in support of the Department of Defense (DoD) initiative in courseware portability. Courseware portability enables interactive courseware (ICW) and associated application programs to operate on delivery systems other than the ones on which they are developed. ICW is a general term used for such applications as computer-assisted instruction, computer-managed instruction, interactive video instruction, and tutorial simulation. The key distinction between interactive courseware and all others is the provision of interactions that tailor, or "individualize," instruction to the needs of individual students. In this way it secures many of the benefits of individualized instruction in group-oriented instructional settings and organizations.

B. APPROACH AND OBJECTIVE

The DoD initiative for ICW portability was undertaken jointly by the Office for Training Policy in the Office of the Assistant Secretary of Defense for Force Management and Personnel and the Defense Audiovisual Policy Office in the Office of the Assistant Secretary of Defense for Public Affairs, American Forces Information Service. These organizations are pursuing a phased approach with an initial objective of system-level portability. The initial objective has been reached largely through specification of a virtual device interface (VDI), an approach that is functional rather than procedural. It specifies a layer of software (the VDI) to be interposed between the application software (the ICW) and the operating system of the computer. The VDI is unique to each operating system and hardware combination, but it is the same for all authoring software for a given class of platforms. It allows any ICW application using the authoring software to operate on any operating system and hardware platform combination that includes the VDI layer, without reprogramming.

C. CURRENT STATUS

The current standard defines the commands, which are issued by the ICW program, and their expected execution, which is controlled by the system hardware devices. The standard is a specification for software. There are two forms of the specification: one uses characters, an ASCII interface, to specify the commands; the other uses numerical tokens, a binary interface, to specify the commands. The ASCII interface is easier to read and understand; the binary interface is more efficient, requiring less memory and eliminating one layer (ASCII to binary) of parameter translation. The current set of commands is organized into service groups for: (1) system management, (2) visual management, (3) videodisc control, and (4) XY-input devices. Other service groups will be added in the near future.

The hardware platform assumed in the current version is an MS-DOS (Version 2.0 or higher) computer with one or more videodisc players, one or more XY devices (e.g., touch screen, mouse, light pen, bit pad), graphics overlay using CGA, EGA, or VGA graphics, conforming system software, and conforming run-time courseware. The requirement for conforming run-time courseware means that authoring software must be modified to issue the standard's interface commands during code generation. The content of most existing interactive courseware will not require modification.

This approach, currently specified in MIL-STD-1379D and DoD Instruction 1322.20, provides for system-level courseware portability. Future work will expand the current approach and standard to: (1) encompass more varieties of ICW, (2) progress from system-level to device-level portability, (3) address new technological opportunities, (4) better address graphics, (5) widen the variety of operating systems covered, (6) progress from platform independence to authoring software independence.

D. BENEFITS

Across many comparisons with conventional instruction, ICW programs have been shown to reduce student instructional time by about 30 percent, increase student achievement by 0.40-0.50 standard deviation units, and cost less than half as much. The DoD portability initiative helps secure these benefits for all Defense training by increasing our capacity to share ICW across a full range of instructional settings within military Services, across military Services, and across internationally allied military Services.

For buyers and users of ICW, portability provides:

- More ICW available off the shelf.
- Increased competition among providers.
- Competition that keys more directly on the costs and instructional effectiveness of the courseware produced instead of its underlying computational requirements.
- Reduced investments of money, manpower, time, and facilities in courseware acquisition.
- Less duplicate funding of course development, since less re-programming will be needed.
- Increased interchangeability, reliability, and maintainability of courseware since its development and production will be more widely standardized.
- A well-defined evolutionary path into an open systems environment.
- Greater preservation of the producers' investments in courseware development over time.
- More flexible accommodation of future technical improvements.
- Improved operational readiness of the Services due to more efficient and effective training and education.

For developers and suppliers of ICW, portability provides:

- A greatly increased marketplace and installed base.
- Access to previously closed markets.
- Reduced development costs.
- A clear path for the evolution of architecture enhancements.
- Reduced stocking and distribution costs.
- Overall increases in the adoption of interactive courseware.

In general, courseware portability will lower the per-unit costs of ICW, lower instructional systems development costs, increase the use of advanced instructional technology in military settings, and increase instructional efficiency in the military Services.

I. INTRODUCTION

Suppose, for the sake of illustration, automobile ownership was complicated by requiring every one of the 40-50 makes of automobiles commonly found on our streets to use a unique type of gasoline. What sort of impact would this state of affairs have?

Certainly, the technical and logistical demands on car buyers and owners would increase. They would have to learn in considerable detail what fuel their cars needed, where appropriate and reliable sources of fuel were located, which brands of fuel for other automobiles they could use in emergencies, and which they should never use. One can imagine user groups growing up around particular automobiles and sponsoring labyrinthine discussions of which fuels could be used after making what technical adjustments in which engines of their automobiles.

The burden on gasoline suppliers would also increase because of complications in arranging fuel deliveries, maintaining adequate supplies, and serving customers who, in turn, would require both training and ongoing technical support to keep their automobiles running. Properly trained gas station attendants would become harder to find and more expensive to hire. Legal issues would certainly arise over which fuels dealers had a legal right to sell to whom under what sorts of licensing agreements. These complications would be solved at considerable cost, which would be passed on to the already beleaguered car owners.

Technical innovation in the development of automobile technology would be hobbled. Developers and suppliers would be discouraged from making technical innovations because research and development investments would be returned only by the owners of one make of automobile. Moreover, many new cost-effective features in both automobiles and their fuels would prove to be incompatible with existing fuel requirements. Policies to ensure compatibility could be established, but these would require increased coordination among manufacturers, who, in turn, would need increased control over the market, again at the expense of consumers, to implement their policies.

In general, the impact on the market for automobiles and fuel would be disastrous. Automobile ownership would be far more expensive, cumbersome, and technically

demanding for everyone involved, and the growth in automobile technology would occur at a rate slow enough to discourage its most enthusiastic supporters.

The analogy to be drawn is obvious. Software is the fuel of our computing technology. Portability, which in this discussion means the ability to operate the same software across many different computer platforms, (1) reduces the costs and increases the usability of both computers and their software, (2) strengthens the market for both, and (3) helps institutionalize their use in many diverse applications.

Without portability, different platforms require different versions of the same application software, and in instructional applications they require different versions of the same interactive courseware. Without portability for the interactive courseware we use in military training and education, routine and widespread use of this promising technology is seriously handicapped. As suggested by the analogy with automobile fuel, these handicaps include additional cost, difficulty of use, technological inertia, and unnecessary complexity in courseware design, development, delivery, and implementation. These handicaps are removed when interactive courseware is designed to be portable.

Interactive courseware can be portable if developers use standard practices to create it. This is to say that portability requires the establishment and the use of standards, and that the topic of portability is also a topic of standards--military standards, government information-processing standards, national standards, and international standards. Implementation of these standards represents a significant opportunity and requirement for cooperation at all levels. This cooperation will improve the quality and reduce the costs of the education and training we provide at all levels of instruction. This paper discusses what is meant by courseware portability, what has been done about it thus far, current and foreseeable benefits, and some recommendations.

II. PORTABILITY

This discussion concerns the portability of interactive courseware (ICW). Interactive courseware is a term that is increasingly used in the United States as a catch-all for such applications as computer-assisted instruction, computer-managed instruction, interactive video instruction, and tutorial simulation. Any of these can be presented by a computer alone, a computer with a videotape or videodisc player, a computer stimulating the actual equipment used on the job, or a computer using compact disc technologies such as Compact Disc Interactive (CDI) and Digital Video Interactive (DVI). As defined by the Department of Defense (DoD), interactive courseware is computer-controlled courseware that responds to individual student input in determining the pace, sequence, and content of instructional presentations. Courseware by itself refers to all training materials, including the curriculum database and all disks, tapes, books, charts, and computer programs, necessary to deliver an interactive courseware program.

The key distinction between an interactive courseware program and all others is the provision of interactions that tailor the instruction to the needs of individual students. The desirability of individualizing instruction has been noted throughout the history of instruction--perhaps as early as the 4th Century B.C. (Corno and Snow, 1986). By tailoring instruction to individual needs, each student receives the level of detail, pace, difficulty, and remediation, and the sequence of topics and interactions needed to learn the material efficiently within the limits imposed by time and access to instructional resources. Interactive courseware programs can provide individualized instruction within our current group-oriented institutions. This advantage, combined with the fact that interactive courseware programs tend to be developed as stand-alone, autonomous modules, make them promising candidates for smooth transport of training across many different application areas, instructional settings, and hardware platforms.

Portability means different things to different people. Transportability, transferability, convertibility, and related terms are used to discuss the same concept. A restricted, but straightforward definition was presented by Dahlstrand in 1984, who defined portability as "the ability to move an application from one computer to another unchanged and get the same results" (p. 17). The DoD definition is not much different.

DoD Instruction 1322.20 defines portability as "The capability to run courseware and associated application programs without modification on a delivery system other than the one for which they were originally designed" (p. 3-1). Interoperability is sometimes used in place of portability, but it generally means portability without re-compilation. Portable applications may or may not have to be re-compiled to run on different systems, interoperable applications do not have to be re-compiled.

A. THE REQUIREMENT FOR PORTABILITY

Over the next several years, the U.S. military Services will invest many millions of dollars to develop interactive courseware. To support this investment, they will acquire a variety of computers, operating systems, peripherals such as videodisc and compact disc units, and all the interface hardware and software needed to communicate within and across these systems. The courseware will be used in all military instructional settings. It will be used by the Reserves and National Guard, in formal courses in military schools, in active duty operational units in garrison and at operational sites, in combined arms and joint staff training, and in educational settings like the Service War Colleges, the Service Academies, and civilian institutions providing Reserve Officer training.

About a dozen different system configurations, at least four different operating systems, and over a dozen authoring systems are commonly used for developing and delivering instruction in the U.S. military Services. Courseware is being produced independently for each of these systems, and in many cases development is proceeding as rapidly as possible to establish a well-anchored position before system standards are introduced. Most of this courseware would have to be "re-purposed", or reprogrammed, to operate on any system platform other than the one for which it was originally developed. Portability will encourage this development to continue while increasing opportunities to share its productions.

Courseware portability will increase the sharing of ICW within each military Service by providing ICW programs that can be used with little, if any, reprogramming across a full range of within-Service instructional settings including initial skill training, advanced skill training, garrison training, job-site training, officer education, and Reserve Component training.

Courseware portability will also increase the sharing of ICW across the different military Services. All the Services teach basic courses such as electricity, electronics, hydraulics, and wheeled vehicle repair. It is not unreasonable to suggest that materials for

these courses, or some portion of them, be usable by all the Services. This sharing is neither an administrative nor political impossibility. Joint training is now offered in 62 inter-Service courses or skill areas (Military Manpower Training Report, 1992), and the need for it will increase as training budgets decrease.

Additionally there are actions being taken to encourage, if not mandate, the DoD to share its instructional materials with organizations and users outside the DoD. The Training Technology Transfer Act enacted by Congress in 1988 is intended to encourage transfer of instructional software from government agencies to non-government activities that support the education, training, and retraining of industrial workers, especially workers in small business concerns. As the major developer and purchaser of instructional courseware in the federal government, the DoD is under increasing pressure to provide for the transfer of its instructional materials to non-DoD users. Fletcher, Bosco, Wienclaw, Ashcraft, and Boycan (1991) have outlined issues and processes involved in such a transfer. Portability is prominent among these issues. It will provide a technical foundation for the DoD to meet the requirements of the Training Technology Transfer Act with minimum impact on the resources needed to perform its operational missions. Other federal agencies that adopt the DoD portability procedures should benefit similarly.

Finally, transfer of DoD materials to military and non-military markets is becoming internationally desirable. The establishment of DoD portability procedures that cross international boundaries will be a significant step in this direction. The possibility of opening international markets to courseware developers should motivate greater investment and development in ICW products and technology, improving their quality, lowering their costs, and thereby benefiting both military and non-military users.

B. THE OPPORTUNITY FOR PORTABILITY

Portability is technically achievable. It is more of an administrative and political issue than a technical issue. As suggested above, portability requires the establishment of policy supported by standards. However, a major issue to be resolved in establishing portability policy is deciding what should be standardized, which is both a technical and an administrative issue. Four possibilities are standards based on (1) hardware, (2) operating systems, (3) authoring software, or (4) the interfaces between these components. Standards based on *hardware* would specify that all interactive courseware use the same generic hardware specified by its operating and performance characteristics. Standards based on *operating systems* would require all interactive courseware to use the same

operating system, again with specified operating and performance characteristics. Standards based on *authoring software* would specify that all interactive courseware use the same authoring software. Standards based on a standard *interface* would establish common techniques for application programs, such as courseware programs and their authoring software, to communicate with and control hardware peripherals.

Despite their conceptual simplicity, there are at least four problems with the first three of these approaches, which would standardize hardware, operating systems, or authoring software. First, all three depend on a common elements specification. Unique features that do not occur in all instances of hardware, operating systems, or authoring software are not likely to be covered by the standard. Unique capabilities in any one of these three areas may languish for many years before being included in the standard, which, in the interim, may be routinely bypassed by exceptions created for users who need access to these capabilities.

Second, changes in the standard must provide upward compatibility for hardware, operating systems, and authoring software already in place. If standards specify hardware, operating systems, or authoring software, then revisions for upward compatibility can accumulate in both quantity and complexity until the standards that incorporate them become unworkable.

Third, the one-size-fits-all philosophy does not work in practice. Different interactive courseware may require different hardware, operating system support, and/or authoring software for good reasons involving both costs and performance. Requiring a single hardware configuration, operating system, or suite of authoring software that is intended to satisfy all users inevitably costs more and satisfies no one.

Fourth, competition to produce quality courseware at reasonable prices must be preserved. Although a standard based on a single hardware configuration, operating system, or suite of authoring software can be expressed in generic terms, it will substantially increase the possibility that military training procurements will be captured by an exclusively small set of hardware and/or software vendors. Acquisitions must be kept as open as possible.

The DoD is pursuing an approach that is based on standardizing communications between application programs and the devices they must control. This approach meets the criteria suggested by the above considerations. It can cover unique elements, it allows upward compatibility and migration to new systems without requiring substantial

modifications in the standard or its structure, it allows developers considerable freedom to exercise their creativity in choosing and developing cost-effective combinations of hardware, software, and operating systems, and, it preserves competition among potential suppliers. Because the interface specified by this standard is generic and intended to apply across many applications and many different physical devices, it is called a virtual device interface (VDI).

C. THE OBJECTIVES OF DoD PORTABILITY INITIATIVES

The current state of affairs and the objectives of the DoD portability effort are illustrated in Figures 1a-c. Figure 1a illustrates the situation without portability. In this situation, application software must be uniquely developed for different operating systems and hardware platforms. The interface between the authoring software and the operating system is unique and usually proprietary for each combination of authoring software, operating system, and hardware platform.

The initial goal for DoD portability is illustrated by Figure 1b. It may be described as system-level portability. Its goal is for application software to run on different operating systems and hardware platforms with no modifications to the application. This goal is rarely reached to the point of requiring no modifications, but a proper scheme of portability minimizes these modifications and facilitates their implementation. The goal has largely been reached by the effort described here through specification of the virtual device interface.

A second, more ambitious objective for portability is illustrated by Figure 1c. This objective may be called device-level, or plug-and-play portability. Its goal is not only for application software to run on different operating systems and hardware platforms with no modifications to the application (system-level portability), but also to permit free substitution of hardware devices, perhaps supplied by different manufacturers, in hardware platforms with no modifications required in either the application or the operating system software. This goal has been achieved elsewhere, for instance by the high fidelity audio community. Records, tapes, and compact discs can be played without modification across a large variety of hardware devices supplied by an equally large variety of hardware manufacturers. Device-level portability has been more difficult to achieve in the computer world, but it too, like system-level portability, can be accomplished through the specification of an interface, in this case a device-handling interface.

Figure 1a. Standard Architecture Without Portability

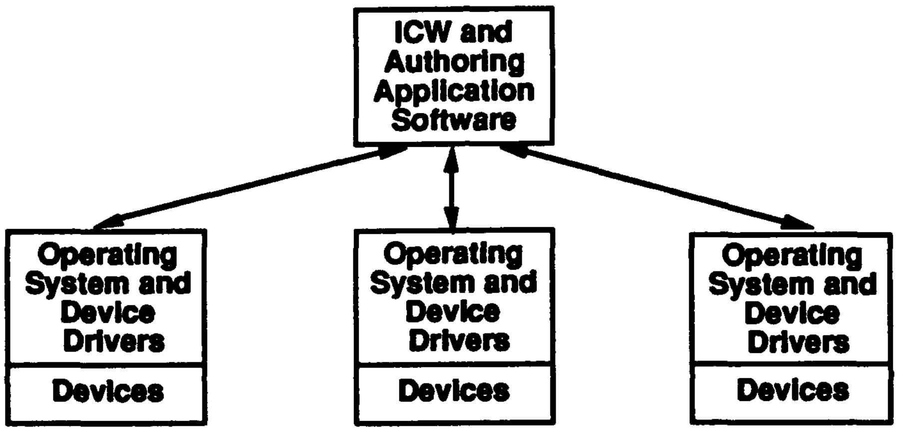

Figure 1b. Standard Architecture with System-Level Portability

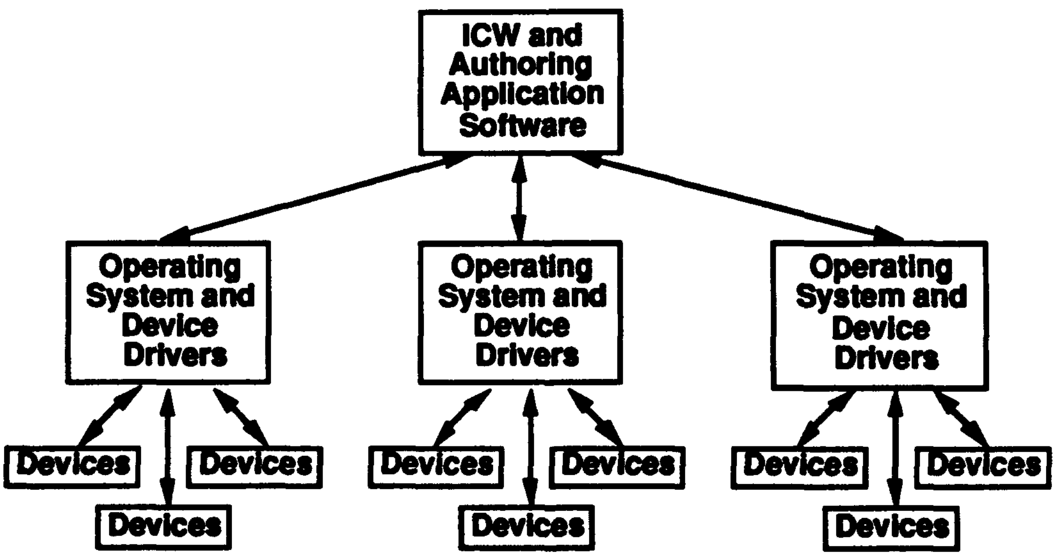

Figure 1c. Standard Architecture with Device-Level Portability

It would have been possible to produce a device-level specification directly without a system-level specification as an intermediate step. There were two reasons for the intermediate strategy chosen. First, the DoD needed to produce a specification quickly, and a system-level specification is quicker and less risky to produce than a device-level specification. Second, a system-level specification is more likely to be widely and easily accepted by the interactive courseware community than a device-level specification. In this sense, the strategy was successful. The specification was produced quickly, and it has met with wide and often enthusiastic acceptance. The risk in the strategy is that the effort may be too successful and fail to progress beyond a system-level specification. Notably, a plug-and-play technical working group has been formed and is scheduled to produce a device-level specification in the next 18 months.

D. TECHNICAL APPROACH

The standard that was produced grew out of close cooperation within the DoD between its Training Policy and Audiovisual Policy Offices. Its development also depended on cooperation with an industrial organization, the Interactive Video Industry Association (IVIA), now called the Interactive Multimedia Association (IMA). These organizations adopted a specific strategy and course of action to establish a system-level virtual device interface standard for the portability of interactive courseware. As Lewis (1991) discusses in more detail, this strategy produced an initial standard that could be quickly developed, easily implemented, and widely accepted. The initial standard was developed by a working committee of the IVIA in cooperation with DoD. Version 1.1 of the standard was published by Dodds, Lewis, McFarling, Mistrot, Snowman, and Spiegelberg in 1990 and was incorporated in Appendix D of MIL-STD-1379D, "Military Training Programs," dated 5 December 1990. Conformance with this standard is required by DoD Instruction 1322.20, "Development and Management of Interactive Courseware (ICW) for Military Training," dated 14 March 1991.

The focus of efforts to accomplish portability has changed in the last few years. Earlier discussions by commentators such as Brown (1977) and Dahlstrand (1984) focused on achieving portability through the use of standard higher order computer languages. Dahlstrand emphasized that portable programs should be in source form and that data files should be in text form (stored as formatted characters). More recently the focus has broadened to include both higher and lower levels of concern. Collis and De Diana (1990) presented a seven-level model of portability factors in the life cycle of interactive

courseware. The model includes higher level technical, educational, social/cultural, and organizational factors as well as lower level factors such as algorithms and data. The virtual device interface approach pursued by the DoD does not address the syntax or form of the higher order languages that might be used for authoring interactive courseware and it does not address the syntax or form of the (presumably) lower order language in which the operating system and its device drivers were written. Instead it aims somewhere in between these two levels by specifying how they should communicate about the specific functions that the lower level provides and the higher level accesses.

Basically, this approach concentrates on *what* we want done rather than *how* the hardware does it; its orientation is functional rather than procedural. In practice, most interactive courseware programs and authoring software address operating system subprograms, generally called "drivers", that provide the desired functionalities and access to hardware devices such as visual displays, videodisc players, monitors, and cursor controllers. If communication between these drivers and the code generated by authoring software can be standardized, we will establish a significant degree of portability.

To see in more detail how this approach works, we need to start with a shared view of the applications and systems software used to support interactive courseware. Figure 2 shows interactive courseware (ICW), authoring software, an operating system, and device drivers as successive layers of a system of software, which is a common way to describe such systems. Three notions are assumed in such a description. First, each layer may address only the layers immediately above and below it. Second, such communication occurs in a clear, well-understood fashion; the interface between any two layers is standardized. Third, software developers are free to pursue whatever approaches they wish within their layer of concern--only the interfaces between layers are constrained by standards.

In this view, everything above the Operating System could be described as an application, or as application software. Figure 2 shows a typical ICW application. It contains:

- An ICW data base, which consists of courseware data such as student and course parameters, questions or other items, text, graphics, audio, video stills, and video segments.
- An ICW management program, which consists of the courseware logic--the software controlling the selection, sequence, style, and content of courseware data presented to the student.

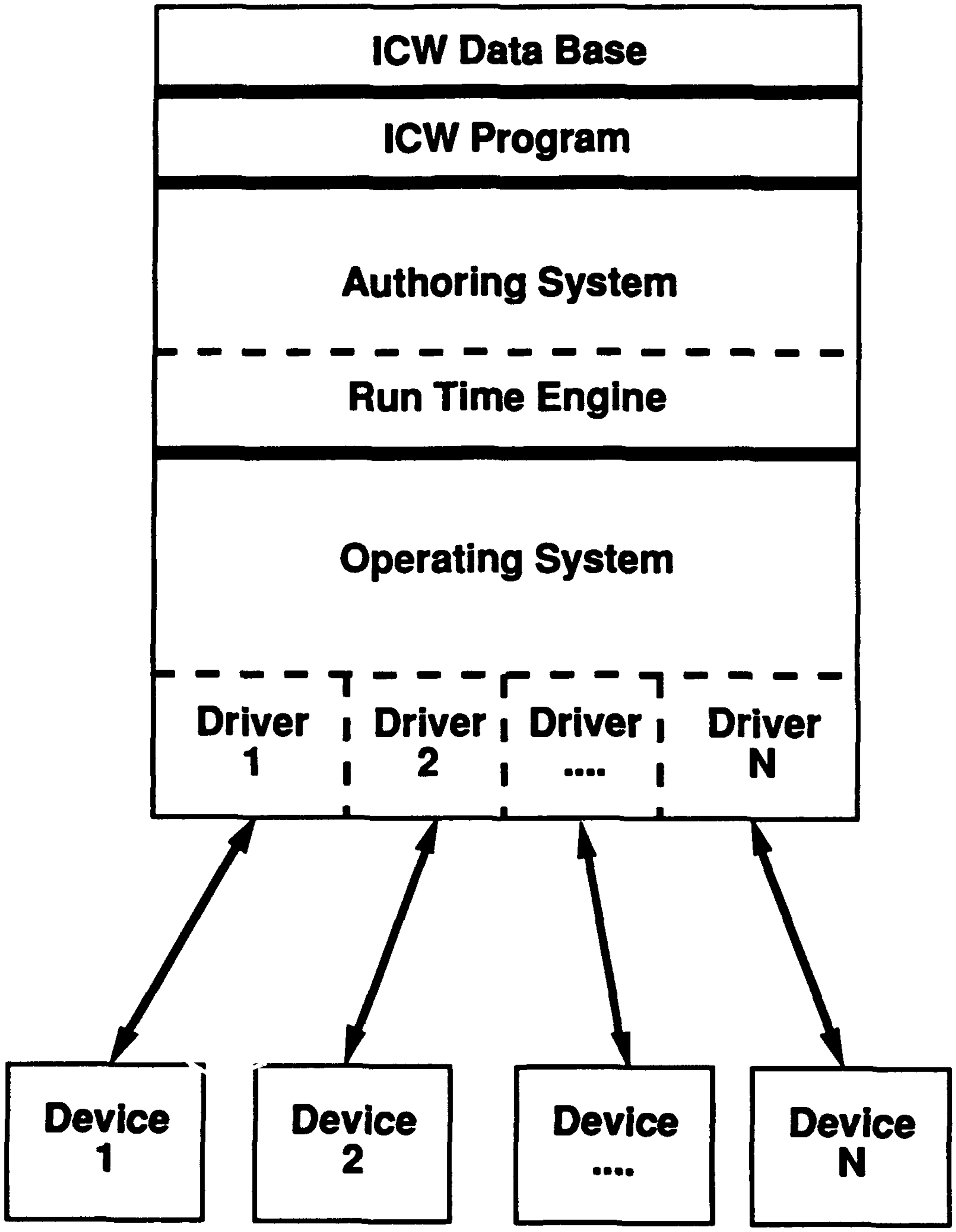

Figure 2. Standard Architecture for Interactive Courseware

- Authoring software, which is used to translate the ICW program and data into a form that can be scheduled and controlled by the operating system and be executed by the hardware. The run time engine is that portion of the authoring software that is needed for execution, but not development, of the ICW program.

As Figure 2 suggests, communications between the authoring software and the operating system are rarely seen or accessed directly by courseware developers. The authoring software generates communications in the appropriate form and these are transmitted across its interface with the operating system when it translates the courseware into statements that are scheduled and executed by the operating system, device drivers, and the hardware at run time. Readers accustomed to MS-DOS operating systems should be advised that the operating system in this case is being treated in a generic manner and that it includes other system services such as the Basic Input/Output System (BIOS), which may reside on chips--in Read-Only Memory--as firmware.

The virtual device interface (VDI) approach standardizes communications between the application software and the operating system with its associated hardware. It is shown in Figure 3, which is the same as Figure 2 except that it interposes a new, small layer of software, called the virtual device interface, between the application software and the operating system. This VDI layer is unique to each operating system and hardware configuration, but it is the same for all authoring software for a given class of platforms. It allows any ICW application using the authoring software to run on any operating system and hardware platform combination that includes the VDI layer, without reprogramming the application. It allows developers flexibility to do whatever ingenuity and imagination permits, and it allows the ICW application to operate wherever the VDI is provided. The costs for this combination of flexibility and simplicity are the development costs to standardize the calls made by the authoring software run-time engine to the VDI, the development costs to write the VDI for the operating system with its hardware configuration, and the run-time costs to service the VDI code at run-time. In experience thus far involving 4 different but already existing ICW authoring systems, we have found the development costs to be 1-4 person weeks, even for poorly designed authoring software (Dodds, personal communication, 19 July 1991). The run-time costs have been negligible and cannot be detected by users.

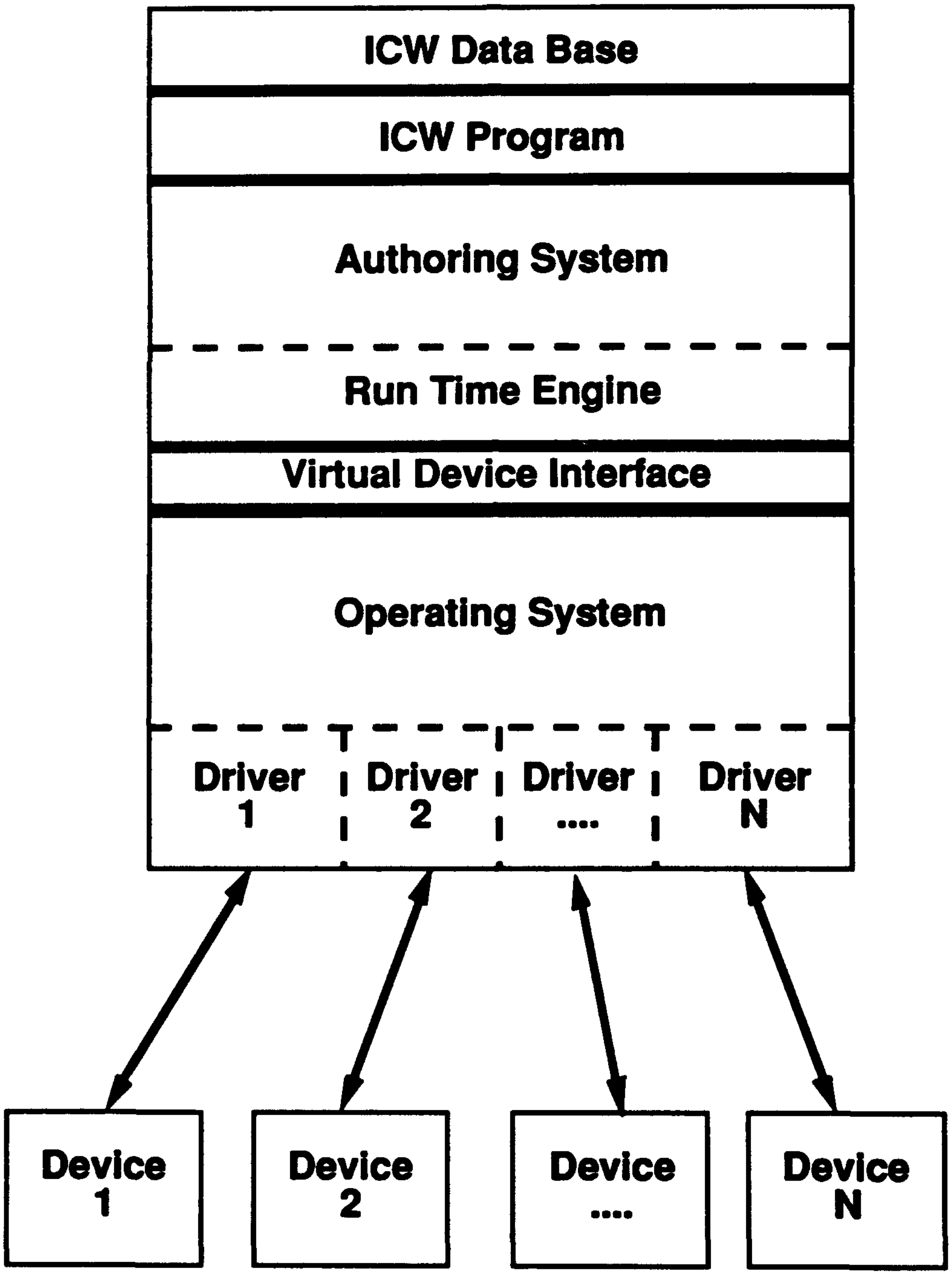

Figure 3. Standard Architecture for System-Level Portability

The lack of a standard, device-level interface for the operating system device drivers is what requires the VDI to be unique for each class of hardware platforms, such as an Intel-based MS-DOS system accessing a specific set of peripheral devices. Figure 4 illustrates an architecture that provides device-level, plug-and-play portability. It incorporates another standard interface, a device-handler interface (DHI) to be interposed between the operating system and its device drivers. In this case the VDIs would be nearly, if not exactly, the same within a more general class of ICW platforms, such as an Intel-based MS-DOS system accessing any set of peripheral devices. The additional costs for this combination of simplicity and flexibility are the development costs to write the DHI, to modify the operating system to include it, and for peripheral device (e.g., videodisc players, light pens, keyboards, monitors) manufacturers to prepare device drivers for the operating system that can communicate successfully with the DHI from below. The run-time cost is just that required to service the DHI code, but as with the VDI overhead, this cost should be negligible.

Work by the DoD on device-level, plug-and-play portability illustrated in Figures 1c and 4 is underway but not yet completed. The system-level portability illustrated in Figures 1b and 3 has been specified and incorporated in a military standard, MIL-STD-1379D, Appendix D, and referenced in a DoD Instruction, DoDI 1322.20.

In brief, the VDI approach is intended to provide the portability needed to satisfy criteria of cost, effectiveness, and control and developers' needs for maximum flexibility in satisfying the requirements of specific instructional applications. It provides a standard way for ICW applications (courseware and authoring software) to interface with the operating system routines that control hardware. Communication from above (from the applications level to the operating-system level) and from below (from the operating-system level back to the applications level) is standardized. Outside of these requirements, all other components of the system can function as usual, unaffected by the standard.

Use of this approach means that most ICW programs already in place do not have to be changed to conform with a virtual device interface standard--code generation for the authoring software may be the only component of the interactive courseware system that needs to be modified. If the authoring software compiles free standing code, the ICW programs may have to be recompiled to conform with the standard, but this should not be a major undertaking, and DoDI 1322.20 establishes procedures to minimize its costs by

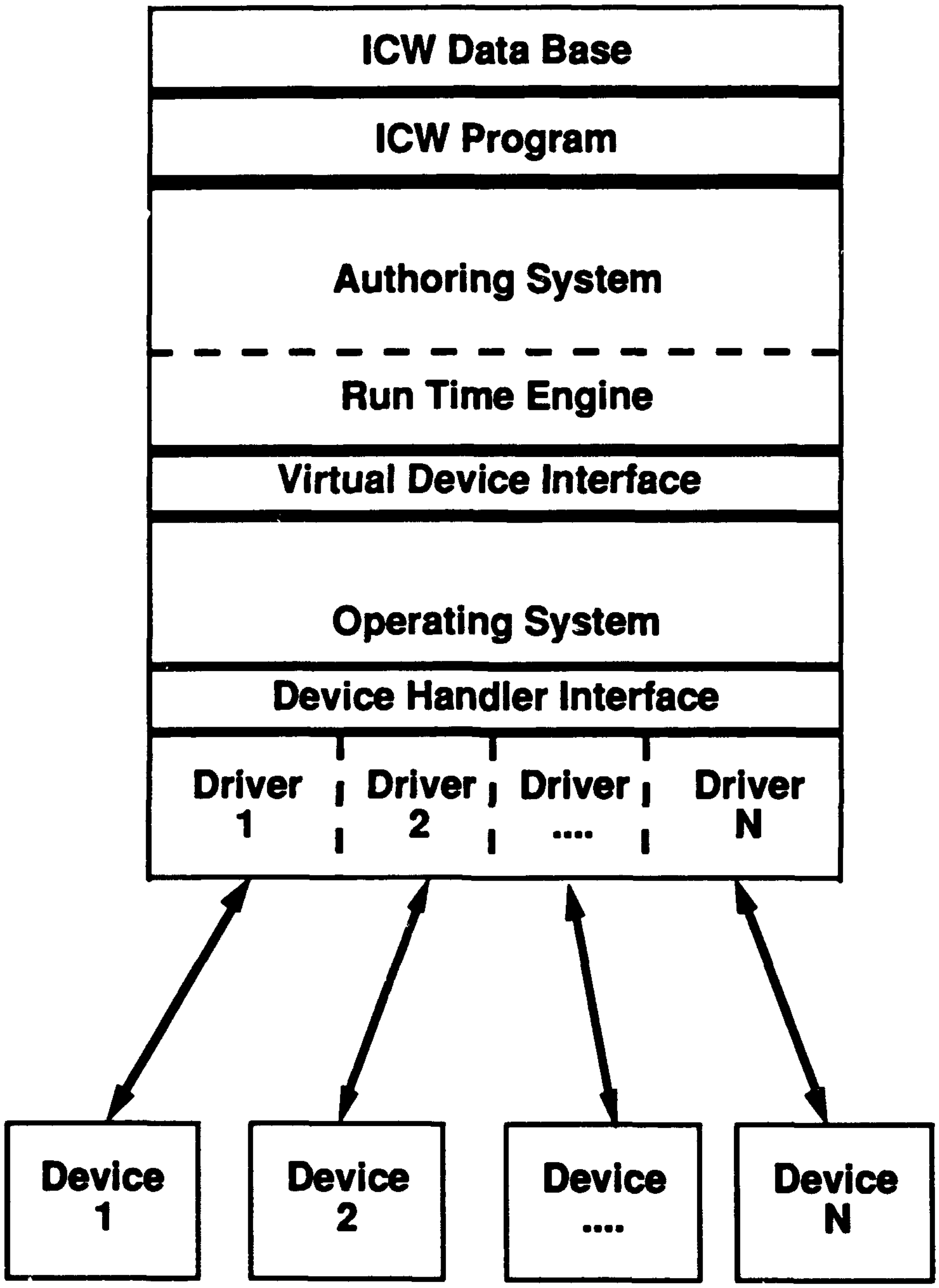

Figure 4. Standard Architecture for Device-Level, Plug-and-Play Portability

requiring that the software needed for compilation be stored and kept available. From a hardware standpoint, no changes in functionality are needed for conformance with the standard.

E. THE CURRENT STANDARD

The current standard defines an interface between application software, such as interactive courseware and authoring software, and operating-system software, which controls access to the resources of the hardware system on which the application software operates. It defines the commands issued by the application software and their expected execution by the hardware, primarily the system peripheral devices. The standard is not itself software, it is a specification for software.

There are two forms of the interface definition. One form uses characters, an ASCII interface; the other uses bits, a binary interface. The ASCII interface uses character strings and file I/O with an MS-DOS installable device driver. Of the two approaches, it is less efficient and requires the procedure parameters used to control devices to be translated to ASCII rather than allow them to be used directly as literal values. However, the ASCII interface is easier to read and understand, it is more like English, and it can be used by any language that can access an MS-DOS file--effectively, any language available on MS-DOS.

The binary interface uses tokens (i.e., numbers) to provide an efficient way to invoke hardware functionalities in MS-DOS using a software interrupt. It is not supported by every language, and it requires more sophistication on the part of its users. However, it is efficient, requires less memory than does the ASCII interface, and eliminates one layer (ASCII to binary) of parameter translation. The two forms differ in user requirements and system performance but are equivalent in purpose and basic functionality.

Seven design criteria were established for the current standard. It was to:

(1) Be compatible with most programming languages.

(2) Function as consistently as possible both for single operating systems and across different operating systems.

(3) Be upwardly compatible to new hardware and new system capabilities so that the standard can be upgraded as required by technological developments without affecting existing applications.

(4) Allow both simple, easy-to-use functions for doing simple tasks and sophisticated functions to support the most demanding multimedia applications.

(5) Provide commands that do not depend on the capabilities of any one operating system.

(6) Not require a specific system software implementation or structure.

(7) Be as inexpensive as possible to implement in terms of memory and performance.

The current standard meets these criteria. It assumes the following as the minimal platform on which it is to be implemented:

- Intel 80X86 architecture.
- MS-DOS version 2.0, or higher, or a functionally equivalent operating system with IBM PC/AT compatible ROM BIOS.
- IBM PC, IBM AT, Microchannel, or Enhanced Industry-Standard Architecture (EISA) bus.
- Graphics and video overlay using CGA, EGA, or VGA graphics, or two separate graphics and video monitors.
- One or more videodisc players or functionally equivalent video sources and one or more XY-input devices (e.g., touch screen, mouse, light pen, bit pad).

This platform meets the requirements of the standard. On a more practical level, a platform that would support ICW programs in a manner that satisfies nearly all performance requirements would be as above but would include such additions and enhancements as the following:

- At least 640 KB of random access memory, a hard disk, and at least one 5.25-inch 1.2 MB or 3-inch 1.44 MB floppy drive.
- MS-DOS version 3.3, or higher, or a functionally equivalent operating system.
- I/O ports as required by the videodisc player and XY-input device, and at least one IBM AT-compatible parallel port.
- Fully VGA compatible graphics with CGA/EGA emulation in overlay modes and with NTSC or PAL overlay capability.
- AT keyboard, at least one button on the mouse, and Laservision compatibility in the videodisc player along with a capability to play either constant angular velocity or constant linear velocity videodiscs.

In short, what is needed now to make existing interactive courseware portable in accord with this system is an MS-DOS computer, CGA, EGA, or VGA graphics, conforming system software, and conforming run-time courseware. The requirement for conforming run-time courseware means that authoring software must be modified to issue

the standard's interface commands during code generation. The content of existing interactive courseware should not require any modification.

One way to determine what functionalities are addressed by any version of the standard is to look at its "service groups". Currently there are four service groups: (1) general system commands; (2) visual management commands; (3) videodisc commands; and (4) XY-input commands. Although the specific commands in each of these groups could be used directly by an ICW program, that is not the intent of the standard. The intent of the standard is that these commands should be issued by conforming authoring software either as ASCII strings or as binary tokens. Authoring software developers are free to present these commands either verbatim or in higher order forms to their users.

The specific commands in the standard are organized and listed under their service groups and briefly described in the following:

(1) General System Service Group. This group is the only one of the four that must be included in all conforming implementations. It concerns general VDI issues and manages information about VDI software, its configuration, and its state. The commands currently in this group are:

syINIT	Initializes VDI management and the sy service group.
sySTOP	Releases all resources used by the interface and makes them available for other uses.
syGETSTATE	Identifies the service groups for which the system was configured, the version of the standard it supports, and other information on the VDI software release.
syCHECKERROR	Returns the number of the last error, if any, and the command that caused it. This command is particularly needed to detect errors that occur later rather than in immediate response to application commands.
syQUEUE	Stores commands in an internal queue for later execution. It is especially useful for collecting commands that have critical timing requirements and must be executed in close sequence. The queue can be turned on and off, cleared, and executed.
syERRORMSG	Returns an upper-case ASCII text error message that describes the error that occurred. (This is an extended, optional command)

The command, syERRORMSG, is the only extended command in the current standard. Others may be added in the future. An extended command is implicitly a candidate for the standard, but not yet included. An application can test for its implementation by issuing the command and checking for the error message. Even if the command is not implemented, a conforming application should be able to process the command to the extent that it issues the correct error message if the command is not implemented. The application program must then provide its own work-around scheme to deal with the absence of the command implementation.

(2) Visual Management Service Group. These commands concern management of the display screen. They control the graphics display, video display, visual signals, video modes, and graphics modes. They distinguish between logical and physical colors and assume some number of logical colors taken from a palette of physical colors. For instance, using a VGA controller a system can display 256 colors simultaneously, and visual management commands therefore assume 256 logical colors for VGA mode. The commands currently in the visual management group are:

vmINIT	Initializes visual management and sets visual management parameters to known values.
vmGETSTATE	Gets information about current state of visual management including current settings of parameters and available resources such as the number of logical colors and number of video sources.
vmSETGRAPHICS	Sets the graphics mode and the position of graphics relative to video displays. This command also controls VGA emulation of CGA and EGA modes.
vmSETPALETTE	Sets the amounts of red, green, and blue components in a specified logical color.
vmGETPALETTE	Returns the amounts of red, green, and blue components in a specified logical color.
vmSETVIDEO	Chooses the video mode (NTSC, PAL, or neither) and the video input source if more than one is available.
vmSETTRANS	Sets logical colors to transparent or opaque and turns physical transparency on and off. The command also enables temporary override of specified transparent colors.

vmFADE	Sets fade and dissolve levels for computer generated (overlay) graphics, video displays, and/or the relative level of one to the other. Specified changes to specified levels of intensity occur over a specified time. Conforming visual management software can be developed for video hardware without fade circuitry as long as an on/off capability is supported by the hardware.

(3) Videodisc Commands. These commands initialize, control, and query the status of videodisc players connected to the system. Current technology uses two types of videodiscs--constant angular velocity and constant linear velocity--which differ in the way information is stored on the videodisc and in the functionalities they provide to the user. Constant angular velocity technology stores the same number of frames on every track of the videodisc. It provides more functions (e.g., still frame, scan and frame search, frame-by-frame step forward and backward) and less storage than constant linear velocity technology. The videodisc commands handle both technologies, and treat constant linear videodisc functions as a subset of constant angular velocity functions. The commands currently in this group are:

vdINIT	Initializes videodisc players and sets parameters for their management by the system. This command must be issued for each videodisc player used by the application.
vdGETSTATE	Returns information about the status of a specified videodisc player. This command can be used for purposes such as determining if the player door is open or closed, the current frame number, the state of background play or scan, if the remote control unit is on or off, the player speed, and if the player is parked or spinning.
vdPLAY	Executes videodisc play sequences. The sequences may include any combination of starting frames, ending frames, chapters, directions, and speeds.
vdSCAN	Starts the videodisc player playing either forward or backward from its present position at its maximum speed until it is interrupted by another command or the player reaches an edge of the videodisc. This command is more likely to be used during development than during routine operation of an application.
vdSEARCH	Causes the videodisc player to turn video off, immediately scan to the specified frame or chapter number, and freeze.
vdSTEP	Causes the videodisc player to move forward or backward one frame without blanking the screen.

vdSTILL — Causes the videodisc player to immediately stop at the current frame. With constant angular velocity videodiscs, video remains visible; with constant linear velocity videodiscs the player automatically blanks video.

vdSET — Sets various videodisc player values such as the default logical player number, state of the audio and video channels, index and chapter number displays, and disc spin/park status.

vdPASSTHRU — Communicates directly with a videodisc player to access player features that are not supported by other commands in this service group. This command should not be used in portable applications. It is provided as a convenience to developers who want to use the standard command set for portable applications and who do not want to switch to a different command set for access to non-portable player features.

(4) **XY-input Device Service Group.** These commands concern XY-input devices (e.g., mouse, touchscreen, light pen, bit pad). They define a uniform way to obtain information from these devices and define coordinate spaces. Many of these devices operate in stream mode, which is to say that they make position and selection information available on a continuous basis. The standard treats all XY-input devices as stream-mode devices with continuously available data. Multiple physical devices may be mapped to a single logical device. The commands currently in this group are:

xyINIT — Initializes XY-input devices and sets parameters for their management by the system. This command must be issued for each XY-input device used by the application.

xyGETSTATE — Returns information about XY-input devices such as the scaling of the coordinate space, what XY-input devices are available, whether their cursors are visible, and the number of buttons they have.

xySET — Scales the XY coordinate space, sets the default input device, turns the cursor on and off, and sets the current XY coordinates

xyGETINPUT — Gets X-position, Y-position, and information on button presses. The standard supports multiple (1-32) button devices, but applications should assume single-button devices for maximum portability. The standard reports only if a button has been pressed; it does not handle touchdown, liftoff, or intensity.

F. FURTHER DEVELOPMENT

Further development will occur in at least six areas. First, the standard will grow to encompass more varieties of interactive courseware. The current standard focuses on interactive videodisc courseware. This focus was chosen because a considerable body of courseware is now being developed for presentation by interactive videodisc systems, and a significant portion of DoD interactive courseware could be covered by the initial standard. Also, many functions of interactive courseware are incorporated in interactive videodisc courseware, so an appreciable portion of the functions that eventually must be covered is already present in the initial standard.

Second, the initial standard will become a device-level, "plug-and-play" standard. The progression from non-portability to system-level portability to plug-and-play portability is shown in Figure 1a-c. The current standard is systems level, as illustrated in Figure 1b. It assumes some specific functionalities, and it provides full platform independence across a variety of delivery systems commonly used in the DoD and elsewhere. However, if a single device in an interactive videodisc system is replaced with another, the current standard cannot guarantee that conforming ICW will continue to operate on the newly configured system. Plug-and-play will allow such replacement, and it will increase the ability of peripheral device suppliers to compete freely in the market.

Third, new technological opportunities will be addressed by the standard. These include digital audio, audio management, digital video, graphical user interfaces, and CD-ROM. These technologies will be addressed in new service groups that are likely to be added in the next 1-3 years. Other new opportunities will continually arise, and provisions have been made by the IMA and the DoD for modifying the standard to address them.

Fourth, the treatment of graphics in the standard will progress beyond its current hardware/firmware level. The current standard cites EGA/CGA/VGA graphics, which is a temporary measure subject to all the problems that accompany standards based on rapidly evolving hardware. Much of the technical work needed to accomplish this expansion of the standard is being completed outside the IMA and DoD portability activities. Many needs for portability may be satisfied by the incorporation in the standard of virtual scalability, which may be accomplished soon and appears to be technically straightforward.

Fifth, the current standard will be expanded to cover a variety of operating systems. The current standard focuses on interactive courseware written for MS-DOS. Again, this is reasonable and timely since most DoD interactive courseware operates on MS-DOS.

However, the final version of the standard will incorporate POSIX, which may become the government's standard operating system, the many versions of MS-DOS, other PC-based operating systems now available to DoD users, and operating systems based on open system architectures (such POSIX) that are now emerging. This expansion and establishment of the necessary migration path have been discussed by Schneeman (1991).

Sixth, the standard may, in the long run, progress from platform independence to authoring software independence. The current standard provides platform independence which means that courseware can be *operated* on a variety of delivery systems--it does not provide courseware that can be *modified* across a variety of systems, which will be possible given authoring software independence. If a courseware developer wants to adopt a graphics display from one courseware package and modify it for another, the standard now in place cannot guarantee that this can be done, although its platform independence provisions will facilitate this process by substantially increasing the amount of material available for modification. However, authoring software independence is still needed and desired by courseware developers, and it should be addressed as the standard evolves.

G. PORTABILITY POLICY

Two key formal actions have been taken to establish the portability standard in the United States Department of Defense. First, the standard was incorporated in a formally established military standard. Military standards are created to establish technical requirements for processes, procedures, practices, and methods. MIL-STD-1379D, which includes the ICW portability standard, is the principal DoD standard for training. It establishes: (1) procedures to follow when developirg training programs, including guidelines for writing contracts and delivery orders for training; (2) requirements for using DoD Computer-aided Acquisition and Logistic Support (CALS) for training documents; and (3) requirements for using the VDI approach described here.

Military standards are intended as tools for military acquisition. Their existence does not imply any requirement that they be used. DoD directives and instructions are expressions of DoD policy and required procedures. The main goal of DoD Instruction 1322.20 is the cost-effective use of interactive courseware for military training, and it applies to all interactive courseware developed by or for the DoD.

The instruction sets five policies for the development and management of DoD interactive courseware programs and materials. They are the following:

(1) ICW programs are to be designed to promote portability, following the standard DoD programming protocols developed under the DoD portability initiative and other technical requirements prescribed in MIL-STD-1379D.

(2) Payment of royalties, recurring license, or run-time fees, use taxes, or similar additional payments for ICW and associated materials developed for or by the DoD are to be eliminated.

(3) The Defense Instructional Technology Information System (DITIS) is established to provide an inventory and maintain a catalog of DoD ICW programs for use by all DoD components.

(4) Reproduction master materials must be archived for the life cycle of each ICW program.

(5) DoD Components must ensure the availability of all materials necessary to modify ICW programs throughout their life cycles.

In cooperation with the DoD, the National Institute for Standards and Technology (NIST) has adopted the initial standard as the foundation for a Request for Architecture issued to solicit suggestions and recommendations from industry for further development. The product of this DoD/NIST effort will be offered for consideration and adoption as a federal, national, and international standard.

Opportunities for international cooperation in the international training and education community are obvious and needed. Technical review of the standard by ICW developers in other countries will determine its feasibility for international adoption and, if it does prove feasible, provide a foundation for its acceptance either specifically or in principle. Such a review will also begin discussions among the military organizations of the participating countries on the technical and administrative means to provide ICW portability and increase the use of this promising new technology for military training and education.

III. BENEFITS

The main benefit of the DoD portability initiative is to substantially increase the availability and use of interactive courseware. Evidence on the effectiveness of this instructional approach has been accumulating for over 30 years. The evidence has been summarized in a series of reviews starting with Vinsonhaler and Bass, who in 1972 reported a median student increase in achievement of about 40 percent for interactive, computer-based approaches compared with more conventional approaches; continuing through the studies of Orlansky and String (1979), who found an overall time savings of 30 percent in the use of these approaches in military training; and most recently found in the meta-analyses of Kulik and his associates. These meta-analyses combine the results of many evaluation studies in a quantitative manner and express their overall findings in standard deviation units. C-L. Kulik and Kulik (1986) found an average increase in student achievement of 0.26 standard deviations for computer-based instruction used in higher education (this result is roughly equivalent to increasing the achievement of 50th percentile students to that of 60th percentile students). C-L. Kulik, Kulik, and Shwalb (1986) found an average increase in student achievement of 0.42 standard deviations for computer-based instruction used in adult education, which is roughly equivalent to an increase in student achievement from the 50th to the 66th percentile.

In a meta-analysis that is directly relevant to the interactive videodisc approaches addressed by the portability initiatives discussed here, Fletcher (1990) found an overall increase in achievement of 0.50 standard deviations (roughly an increase from 50th percentile to 69th percentile achievement) for students in military training, industrial training, and higher education associated with the use of interactive videodisc instruction. Fletcher also found that across the 13 cost ratios reported in the studies covered by his review, the average ratio of interactive videodisc instruction costs over the costs for more conventional instruction was 0.36. These findings of greater effectiveness with lower costs suggest that strong cost-effectiveness arguments for using interactive videodisc instruction may exist, but direct experimental examination of this possibility is needed before it can be viewed as conclusive.

It is also interesting to note how much interactive courseware may be available for transfer, provided that it can be made portable. Although the Defense Instructional Technology Information System (DITIS), established by DoD Instruction 1322.20, is only a year old, it now lists 4644 ICW programs that are being used by the military Services. Only computer-based instruction and interactive videodisc have been reported to DITIS as delivery media for ICW programs. Of these programs, 3499 were reported as being computer-based instruction, and 965 were reported as interactive videodisc programs--leaving 180 programs for which the delivery medium remains unreported.

Fletcher, Wienclaw, Bosco, and O'Neil (1992) discussed the number of these ICW programs that may be suitable for transfer to civilian use. They reported potentially transferable ICW programs in four general categories of instruction: (1) Basic skills training and general education (e.g., physical sciences, social sciences, foreign languages); (2) Specific technical training (automotive mechanics, data communications, hydraulics); (3) Workplace knowledge and skills (e.g., document control, employee relations, bookkeeping); and (4) Professional and para-professional knowledge and skills (e.g., engineering, nursing, veterinary medicine). They also categorized the ICW programs under one of three delivery systems: (1) MS-DOS programs using computer-based instruction for their instructional presentations; (2) NOS programs using computer-based instruction for their instructional presentations (developed using the TUTOR authoring language); and (3) MS-DOS programs that include use of interactive videodisc (IVD) for instructional presentations. The results of this analysis are reported in Table 1, which shows that of the 4644 ICW programs now listed in DITIS, 2718 may be candidates for transfer to the private sector.

Table 1. Counts of ICW Programs That Are Candidates for Transfer to the Private Sector[a]

	MS-DOS	NOS	IVD	Total
Basic Skills/Education	41	778	16	835
Technical Training	391	600	69	1060
Workplace Skills	42	227	1	270
Professional Training	553	392	44	392
Totals	591	1997	130	2718

[a] From Fletcher et al. (1992).

This preliminary survey of the DoD inventory suggests that there are ICW programs that are candidates for transfer within the Services, across the Services, from DoD to non-DoD organizations, and perhaps across internationally allied military Services. More comprehensive analyses of current DoD interactive courseware programs and the cost savings of portability remain to be completed. Precise data will emerge as the DITIS data base is filled and as the ICW community gains more experience with the portability standards now in place. However, if the costs of re-programming even a few ICW programs can be avoided, the costs of developing and implementing courseware portability will be recovered.

The issue is not just one of costs. Promotion of ICW and wider realization of its benefits across the DoD are likely to be a significant contribution of the portability standard. Portability should provide more efficient use of the resources that we allocate to military education and training. To the extent that these programs also produce improved performance by students, ICW may increase the quality and quantity of human performance, which is an essential and inseparable component of all DoD activities.

Overall, portability will provide to buyers and users of ICW:

- More ICW available off the shelf.
- Increased competition among providers and competition that keys more directly on the costs and instructional effectiveness of the courseware produced instead of its underlying computational requirements.
- Reduced investments of money, manpower, time, and facilities in courseware acquisition. More courseware will be available and it will be more widely useable.
- Less duplicate funding of course development, since less re-programming will be needed.
- Increased interchangeability, reliability, and maintainability of courseware since its development and production will be more widely standardized.
- A well-defined evolutionary path into an open systems environment.
- Greater preservation of the producers' investments in courseware development over time.
- More flexible accommodation of future technical improvements.
- Improved operational readiness of the Services due to more efficient and effective training and education.

Portability will provide to developers and suppliers of ICW:

- An increased marketplace and installed base.
- Access to previously closed markets.
- Reduced development costs.
- A clear path for the evolution of architecture enhancements.
- Reduced stocking and distribution costs.
- Overall increases in the adoption of interactive courseware.

And, in general, portability will:

- Lower the per-unit costs of ICW.
- Lower instructional systems procurement costs.
- Increase the use of advanced instructional technologies.
- Increase training efficiency.

IV. CURRENT ISSUES

Lecarme, Pellissier-Gart, and Gart observed that "To suppose that a program can carry on an interactive dialog with a terminal is often the surest way to make it unportable" (1989, p. 16), and the defining core of ICW is its interaction with students. In attempting to ensure the portability of ICW, we have not chosen a simple task. Substantive issues remain to be settled. Among them are the following:

- *Continued support.* We have discussed establishing a portability standard in the same way we might discuss building a bridge, as if once in place and given some maintenance, it would continue to perform its function indefinitely. Such is not the case. A standard must not only be sustained but continually improved, reshaped, and reconsidered from newer perspectives. The portability standard is a process as well as a specification, and the process must be maintained. Development work may to a significant degree be carried out by the IMA, and NIST efforts may produce the necessary committees and coordination, but the process will continue to need advocacy and support from the DoD. Resources and responsibilities for this sustained effort are now only loosely established, and the systematic analyses of costs and effectiveness needed to justify advocacy for portability are far from complete.

- *Promulgation.* The standard, which began largely under urging and support by DoD has grown well beyond DoD applications. The DoD is still a major developer and user of ICW, but it is only one participant in what has become a large market. The IMA portability effort is now supported by nearly all the major suppliers of interactive multimedia hardware, software, and courseware. The problem is not widespread acceptance, but the opposite. Because the new participants have little vested interest in the current version of the standard, which is incorporated in MIL-STD-1379D, new versions of the standard may leap so far ahead that the current version, and the ICW programs that conformed to it, will be left without a practicable migration path to follow.

- *Certification.* The portability standard will have little effect if developers and users who wish to conform to the specifications and provisions of the standard have no means for determining if they or their suppliers and contractors have successfully done so. Certification is needed, which in turn means that a resourced facility with clearly defined responsibilities for certification must be established and maintained. This facility does not yet exist; the scope and

nature of its responsibilities have not been determined (for instance, do we need certification of hardware, operating systems, authoring software, courseware, or some combination of all of these), and the technical procedures needed for certification have yet to be devised.

- *ICW hand-off.* A key objective for the effort to establish courseware portability was to increase the sharing of instructional materials within the military Services, across the military Services, and between military and non-military organizations both inside and outside the government. Helping to ensure the technical feasibility of sharing instructional materials is an important component, but technical feasibility will be of little use if the materials themselves cannot be obtained. An organization is needed to establish and maintain an inventory of portable ICW programs. The materials will have to be physically stored with all necessary environmental safeguards. They must be maintained in good working order, and updated and modified promptly by the organization responsible for marketing them. Software, hardware, and an in-house technical capability must be available in order to provide this quality assurance. Users will also require technical assistance in installing the materials for their own applications, and this assistance will have to be provided via telephone hot lines, technical bulletins, user group seminars, and the like. An organization to perform these functions could be established and would probably become self-supporting, but the work to do so needs to be undertaken.

V. DISCUSSION

The portability of interactive courseware is primarily an issue of software. At the system level discussed here, portability is mostly concerned with code generation and the run time engine of authoring software. Portability is an issue that individuals concerned with instructional technology must address, but it is not an issue of instructional content or instructional design. It is an instructional production issue, which is best associated with the production and delivery component of a systems approach to instruction.

Provisions for ICW portability will have a substantial impact, both direct and indirect, on the costs to develop and use ICW in military and non-military applications. By widening the market for ICW and significantly increasing its potential life-cycle return on investment to developers and users, it should both decrease the per unit costs of ICW and promote its widespread use. If early indications of ICW cost-effectiveness turn out to reflect genuine and substantial efficiencies in instruction, ICW portability will increase the quality and quantity of human performance available to our military Services and thereby contribute substantially to their readiness and effectiveness.

Portability does not come free. There are development costs to be borne by suppliers of authoring software systems and, once device-level portability is established, by suppliers of ICW peripheral devices. There are also run-time processing and memory costs of ICW platforms that support the virtual device and device-handling interfaces required by system-level and device-level portability, respectively, as described in this paper. However, these costs are likely to be minor in proportion to the development and run-time costs that are already invested to support ICW and trivial compared to the cost avoidances that will result once conformance with the portability standards is established.

The costs to maintain and further develop the ICW portability standards described here are a more serious matter than the development and run-time costs that suppliers will have to bear. The early DoD initiative to establish ICW portability has been turned over to the Interactive Multimedia Association and its participating members. They are likely to support further development of the standard for the near future. If the National Institute for Standards and Technology (NIST) is successful in fusing the IMA and DoD products

into a Federal Information Processing Standard and later into a standard supported by the American National Standards Institute and the International Standards Organization, its long-term survival will be assured. This process has grown beyond direct DoD influence and support. The key to successful development of courseware portability in DoD is to encourage those aspects of the standards process that directly support DoD training and audiovisual policies and to ensure that new developments ensure graceful migration of DoD products to newer versions of the standard.

There are opportunities for international cooperation in the establishment of ICW portability for military training and education. Review of the work completed thus far to determine the feasibility and desirability of developing international standards for portability would be appropriate and helpful at this time. The most effective approach for undertaking these reviews may be to attempt implementations of the standard in existing and new ICW programs. This hands-on approach with systematic records kept of the time, costs, and critical issues required for these implementations would advance the current state of the art. A central purpose of this paper is to encourage relevant and interested organizations to undertake these reviews.

REFERENCES

Brown, P.J., *Software Portability*, Cambridge University Press, Cambridge, United Kingdom, 1977.

Collis, B.A., and I. De Diana, "The impact of different portability factors during the life cycle of an educational software adaptation project," *Journal of Research on Computing in Education*, *23*, pp. 306-317, 1990.

Corno, L., and R.E. Snow, "Adapting teaching to individual differences among learners," in M.C. Wittrock (Ed.), *Handbook of Research on Teaching (Third Edition)*, Macmillan Publishing, New York, N.Y., 1986.

Dahlstrand, I., *Software Portability and Standards*, Ellis Horwood, Ltd., Chichester, United Kingdom, 1984.

Dodds, P., S. Lewis, D. McFarling, H. Mistrot, G. Snowman, and J. Spiegelberg, *Recommended Practices for Multimedia Portability, Release R 1.1*, The Interactive Multimedia Association, Washington, D.C., 1990.

Fletcher, J.D., *Effectiveness and Cost of Interactive Videodisc Instruction in Defense Training and Education*, IDA Paper P-2372, Institute for Defense Analyses, Alexandria, Va., 1990.

Fletcher, J.D., J. Bosco, R. Wienclaw, J.K. Ashcraft, and G. Boycan, *Assessment of Department of Defense Job Skill Enhancement Programs*, IDA Document D-939, Institute for Defense Analyses, Alexandria, Va., 1991.

Fletcher, J.D., R. Wienclaw, J. Bosco, and H.F. O'Neil, Jr., *Defense Workforce Training Programs*, IDA Paper P-2743, Institute for Defense Analyses, Alexandria, Va., 1992 (in Draft).

Kulik, C-L.C., and J.A. Kulik, "Effectiveness of Computer-Based Education in Colleges," *AEDS Journal*, *19*, pp. 81-108, 1986.

Kulik, C-L.C., J.A. Kulik, and B. Shwalb, "Effectiveness of computer-based instruction in adult education," *Journal of Educational Computing Research*, *2*, pp. 235-252, 1986.

Lecarme, O., M. Pellissier-Gart, and M. Gart, *Software Portability with Microcomputer Issues*, McGraw-Hill, New York, N.Y., 1989.

Lewis, D.S., "Introducing the IMA recommended practices for multimedia portability," in P.V.W. Dodds (Ed.), *IMA Compatibility Project Proceedings*, The Interactive Multimedia Association, Washington, D.C., 1991.

Orlansky, J., and J. String, *Cost effectiveness of computer-based instruction in military training*, IDA Paper P-1375, Institute for Defense Analyses, Arlington, Va., 1977.

Schneeman, R.D., "Interactive video software portability: Migrating an MS-DOS application to POSIX," in *Proceedings of the 29th Annual Technical Symposium, Advanced Information Interfaces: Making Data Accessible*, National Institute of Standards and Technology and Washington, D.C., Chapter of the Association for Computing Machinery, Gaithersburg, Md., 1991.

U.S. Department of Defense, *Military Manpower Training Report: FY 1993*, U.S. Government Printing Office, Washington, D.C., 1992.

U.S. Department of Defense, *Military Standard 1379: Military Training Programs*, U.S. Government Printing Office, Washington, D.C., 1990.

U.S. Department of Defense, *DoD Instruction 1322.20: Development and Management of Interactive Courseware (ICW) for Military Training*, U.S. Government Printing Office, Washington, D.C., 1991.

Vinsonhaler, J.F., and R.K. Bass, "A summary of ten major studies on CAI drill and practice," *Educational Technology, 12*, pp. 29-32, 1972.